C++ Strings

Strings in C++ are very similar to strings in Python. The only major difference is that Python strings are immutable, meaning that once a string is created, you cannot alter the individual characters that comprise it (you can assign a new string to an existing string variable, but that does not modify the first string). C++ strings are mutable, in that you can alter individual characters without re-assigning the whole string. C++ strings, like C++ vectors, don’t do bounds checking: they will not prevent you from reading or writing off the end of a string.

To use these string operations, add `#include <string>` to the top of your code.

String operations

Create a new, empty string	`string s;`
Create a new string and initialize it	`string s = “whatever”;`

Access a character in a string
Use brackets like you do in Python, in all the same places you would in Python, but no slicing or negative indices.
	`s[pos]`
	`(0 <= pos <= s.length() – 1)`

Change a character in a string	`s[pos] = ‘c’;` (Notice that single characters in C++ are enclosed in single quotes; strings use double quotes.)
Access the first character in a string	`s.front()`
Access the last character in a string	`s.back()`
Access the length of a string	`s.size()` or `s.length()`
Make an existing string empty	`s.clear()` [changes s]
Insert a string t into an existing string s	`s.insert(pos, t);` [changes s]
Remove characters from a string (Removes chars at `pos` through pos+count-1, inclusive)	`s.erase(pos, count)` [changes s]
String concatenation	`s = t + u;` [t and u can be strings or characters] `s += t;` [t can be a string or a character] `s.push_back(character);` [changes s] `s.append(string);`
Remove the last character from a string	`s.pop_back()` [changes s]
Access a substring of a string (Returns a new string consisting of characters at `pos` through pos+count-1, inclusive)	`s.substr(pos, count)`
Replace a substring with another string (Removes substr(pos, count) and inserts string t)	`s.replace(pos, count, t);` [changes s]
Find a substring within a string (Returns the left (or right)-most position in s where t is found, starting at position `pos`, -1 if not found)	`s.find(t, pos)` [pos is optional] `s.rfind(t, pos)`

Find one of a set of characters within a string
(Returns the left (or right)-most position in s where any character in t is found, starting at position `pos`, -1 if not found)
	`s.find_first_of(t, pos)`
	`s.find_last_of(t, pos)`

String comparisons	`<  <=  >  >=  ==  !=`
String input and output	`cout << s;` `cin >> s;` [reads until first space] `getline(cin, s);` [reads whole line]
Convert a string to a numeric type (i=int, l=long, ll=long long, f=float, d=double, ld=long double)	`stoi, stol, stoll, stof, stod, stold`
Check if a *character* is uppercase/lowercase/etc (#include <cctype> to use these)	`isupper, islower, isspace, ispunct, isblank`
Convert a *character* to uppercase/lowercase (#include <cctype> to use these)	`tolower, toupper`